Forsch Ingenieurwes (2019) 83:655–668
https://doi.org/10.1007/s10010-019-00322-z

ORIGINALARBEITEN/ORIGINALS

Automated requirements-driven design synthesis of gearboxes with graph-based design languages using state of the art tools

Kevin Holder[1] · **Stephan Rudolph**[2] · **Ralf Stetter**[3] · **Corinna Salander**[2]

Received: 3 April 2019 / Accepted: 24 May 2019

Abstract

In order to best match individual customer requirements in gearbox applications, customer-tailored gearboxes are a desirable solution. That leads to the demand for a truly individualized product development. In order to keep costs down despite an increasing product variability, novel digital design methods which automate the design process and avoid redundant manual work need to be developed and deployed. Graph-based design languages in UML (Unified Modeling Language) use a central data model to guarantee model consistency and to generate automatically multi-disciplinary models and simulations thereof.

The paper shows the use of graph-based design languages for automated gear synthesis and the automated three dimensional arrangement of gearset parts. The aim of this language is to generate the gearset including the gear calculation. The calculation automation is supported with the domain-specific synthesis tool GAP ('Getriebe Auslegungs Programm'). The resulting gearset is then packaged into a given design space provided as one of the customer requirements. The package problem is the subject of several boundary conditions and for this reasons normally multiple solutions do exist. Therefore, optimization methods are used in the following to select the final three-dimensional arrangement. The current state of the gear system model also includes a fully parametric CAD housing.

The approach shown in this paper can be extended to the entire product life cycle, also including domain-specific and already established software solutions. Finally, the opportunity is present to include further product assessments like a cost calculation and a first reliability check of the system.

The authors are aware of the fact that this fully automated process is currently still restricted to a specific class of product design problems such as gears and gearbox design.

Keywords Graph based Design Language · Knowledge based Design · Design Automation · Gear synthesis · Drive train · 3D speed-reducer · Packaging gear systems

1 Introduction

As in almost every area of mechanical engineering, a prominent need can be identified for the development of gear systems for ever shorter development times with the simultaneous goal of reduced development costs. Additionally, the ever more individual customer wishes in ever shorter time need to be met.

According to the approach of Simultaneous Engineering (SE), for example, in the development of gear systems for electric rail vehicles (Electric Multiple Units, EMU), a large number of people with subject-specific knowledge and different software tools are involved sequentially and partly also in parallel for the solution of a multitude of tasks. The exchange of information between people and software tools

✉ Kevin Holder
kevin.holder@zf.com

Stephan Rudolph
stephan.rudolph@ifb.uni-stuttgart.de

Ralf Stetter
ralf.stetter@hs-weingarten.de

Corinna Salander
corinna.salander@ima.uni-stuttgart.de

[1] ZF Friedrichshafen AG, 88046 Friedrichshafen, Germany

[2] Universität Stuttgart, 70569 Stuttgart, Germany

[3] Hochschule Ravensburg-Weingarten, 88250 Weingarten, Germany

is usually already realized on a digital basis[1], but still often with manual interaction, i.e. the information exchange between the individual software tools has to be done by qualified engineers. Especially with the application of simultaneous engineering, this leads to the need for manual data-update interactions several times during the development process.

Due to the increasing product complexity with ever shorter development times, the necessity increases to get rid of steps, which are error-prone and may hinder the actual development progress. A novel, language-based option for significantly increasing the efficiency[2] and effectiveness[3] is the introduction and application of a central data model for multiple domains, which may be provided automatically using a graph-based design language (GBDL) based on UML (Unified Modeling Language) and a design compiler used for its machine translation.

This paper shows how a graph-based design language may be designed for the early stage of the product development process (PDP). Therefore section 1.1 describes the early stage of the PDP for EMU gear systems with its impediments as well as the vision of a design system using GBDLs. Section 1.2 explains the basic concept of GBDLs. The relevant literature of each possible domain is presented and discussed in section 2. Section 3 describes the implemented process with its data model (section 3.2) as well as the spatial arrangement logic for the gear parts (section 3.3.1) and geometic construction of parts like housing and gearset parts (section 3.4). Based on this, section 4 presents the results of an application on real and test scenarios (section 4.1 and section 4.2). The last section 5 expounds further possibilities of the GBDL approach for gear systems. In principle there is the opportunity to include any further product assessments. At the moment a cost calculation and a first reliability check by means of a Fault Tree Analysis (FTA) of the gear stages are under development.

1.1 Motivation

This section describes the motivation, which lead to the research described in this paper. For powertrains consisting of an electric machine and a gearbox, the following sections show a generic and thus very flexible method to support the product development process (PDP), especially in the early phase of order initiation and the subsequent phases of product development.

Numerous programs, which perform calculations and simulations, are used during the PDP of the precedently mentioned transmission gears. For instance, a driving performance simulation for the entire vehicle or the working machines can be carried out. Followed by the determination of gear geometry according to relevant standards [13] in order to meet the lifetime requirements. The use of CAD and the creation of a 3D model does not represent the end of the tool chain, the design must still be verified by FEM calculation and, for example, by analytical calculations for screws and shafts. The individual steps are not necessarily serialized and iterative approximation to a satisfactory design for all involved stakeholders (i.e., customer, supplier, developer, etc.) is currently the common practice.

Although in practice, the sequence of development tools is accelerated as much as possible and the mutual integration of design functions is steadily improved, holistic approaches and truly integrated process chains in development tools are currently not yet applied[4]. A possible reason for this may be the complex implementation, as well as the lack of advantages with only a single break in the entire process chain. Another reason may be the lack of understanding of the advantages of an integrated process.

With the emergence of graph-based design languages, a novel, language-based approach is in place that seems to bring the seamless integration of domains and the resulting holistic approach, including multi-criteria optimization, within reach. It is particularly advantageous to use a graph-based design language when many different individual products have to be designed, but whose components belong to a common design space. This means that for products whose design process is always similar or the same, but by quantitatively different performance data, a need for a new product design paradigm is obvious. For EMU transmission, this situation exists because of the lack of effective standards for vendor-independent bogie platforms. Thus, vehicle manufacturers (OEMs) develop and deploy different transmissions. Although the technology and the structure of the gears is often similar or even the same (e.g. 1-stage helical gear, 2-stage helical gear, 1-stage bevel gear, etc.), the performance data such as gear ratio, speed, torque or geometric dimensions differ very clearly.

In general, a distinction is made between the 'planning phase', the so-called 'offer to order' process, in which order completion is to be supported and promoted, and the series development, which leads to a producible and thus documented overall product design. During the first implementation, the focus is on the project engineering phase of the electric powertrain.

---

[1] mostly employing open, standardized or proprietary data formats in files or databases

[2] Efficiency in the sense of 'doing the right thing'

[3] Effectiveness in the sense of 'doing the thing right'

[4] Software manufacturers like Dassault begin to offer such end-to-end process chains with platforms such as 3D Experience. However, the proprietary data format is not disclosed to the customer, so it remains difficult to exchange data with software systems of other manufacturers outside the limited number of standardized data formats (e.g., STEP, DXF, STL, etc.).

The aim of the current implementation is the modeling of the following subareas into a common program system:

- Illustration of requirements and usecases,
- Dimensioning of gearset and validation (wheels, shafts, bearings),
- Housing design,
- Evaluation and feedback for optimization in the context of the overall model.

1.2 Graph-based Design Language

Graph-based design languages based on UML (Unified Modeling Language) represent an approach for the holistic creation and digital modeling of product designs. In design languages, the individual terms (i.e. the *vocabulary*) represent reusable and relatively freely (re-)combinable language modules. The assembly knowledge about the individual vocabulary (i.e. the *rules*) are mapped as language operations, see left side Fig. 1. Graphs are understood to be the digital information representation in the context of graph-based design languages, the nodes of the graph are abstract placeholders for real objects, processes, or states, while the edges of the graph are relations or information flows between the nodes. The design graph gets expanded during the execution of the implemented model.

Modeling and information processing in a graph-based design language initially occurs completely independently from any particular program-specific or proprietary data format throughout the product life-cycle, using the unified modeling language (UML) which is free, open-source, and internationally standardized and derived from software engineering. Among other things, the product design can be illustrated with the required design knowledge, including different domains, such as the relevant product requirements, design parameters and interfaces.

Until the last steps, the information is processed on an abstract level by means of model transformations (i.e. the *rules*). Only in the last steps, the holistic data model in UML will be transformed into domain-specific language representations (DSLs), see right side Fig. 1. So there is a clear distinction between the representation of knowledge in the form of the object-oriented language definition in UML and the possible further processing in the different process chains and data formats of the different manufacturers [23].

Of course, due to this deliberately chosen abstract knowledge representation, abstractly formulated vocabulary and rules must be found, written down and processed in a language compiler for all terms occurring along the product life cycle and their interactions. The required abstract representation of a concrete CAD product geometry in the form of a so-called ‘abstract geometry’ (in UML) and for further multi-physical product properties (e.g. in the form of MKS, FEM and CFD models for multi-body-simulation, structural-mechanical and fluid-mechanical simulation) in the form of a so-called ‘abstract physics’ (also in UML)

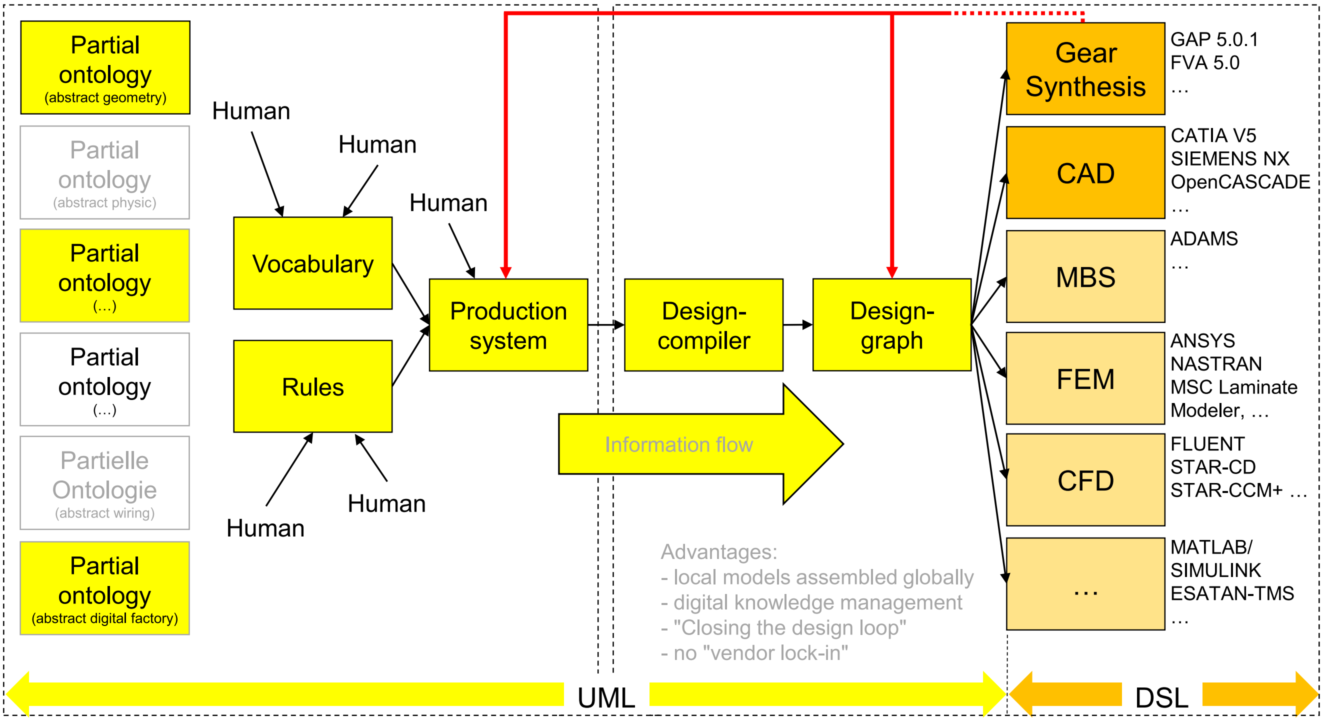

Fig. 1 Information flow in graph-based design languages based on [25]

in graph-based design languages, allows the information representation in UML independent of a specific data format in a special domain or from a specific vendor-specific or proprietary data formats. The final transformation from UML into one or more target languages (Domain-Specific Languages, DSLs) takes place later and through a further machine processing step in the language compiler or its plug-ins [29].

2 State of the art

This section describes the current state of the art in the area of digital gear system development. Digital modeling in product design is already well researched. In addition to generally applicable CAD systems, program systems are also used for digital modeling of gear systems[5], which are specially designed to support gear design processes. Present in industry are e.g. software tools such as the FVA Workbench [8], KISSsoft [17] respecitvely KISSsys [18] or also company owned software solutions for the design and especially recalculation of gears in gearboxes and drive trains. Also for the other machine elements in gearboxes such as shafts and bearings there are specialized tools that perform calculations according to standards or specific company internal knowledge. For bearings Bearinx [27] is frequently employed. The mentioned tools are partly not only specialized in one or a few machine elements, but there are already approaches to model gears as an entire system. The FVA Workbench can also take into account couplings, shifting positions (and therefore different power flows) as well as housing deformations.

Usually gears are designed iteratively, based on experience or by standards [5] an initial concept is defined, which can then be assessed by standards regarding its suitability. The calculated safety against pitting and tooth fatigue fracture according to [14] can then be used for an adaptation of the initial or previous concept. In the publications [6], [3] and [20] a method is proposed, which allows an explicit dimensioning and design of gear teeth, i.e. it leads over the proposition of safety margins and the performance data to a possible validated geometry. This approach is implemented in GAP (abbreviation 'Getriebe Auslegungs Programm'). This kind of approach appears to be particularly suitable for automation because the process does not require human intervention during the iteration and a programming interface is present (API) [21].

In the development of gear systems, the performance requirements, which describe the characteristics for torque and speed conversion, still add spatial requirements. It may be a task to transfer power from one place in space to another place, or to realize a gear system, which is as compact as possible. The solution of the package problem of a transmission or powertrain is addressed in [26] and [2]. In the publication [7] a method is proposed, with which gear set structures can be placed automatically in a given space and the power path of the gear system in space is kept as compact as possible.

The STEP format is discussed as exchange format for gear systems in [16]. With REXS (Reusable Engineering Exchange Standard) [9] proposed an open general format for the exchange of transmission model data based on XML [32]. The specification currently includes the following machine elements: shafts, spur gears and gears, bevel gear stages and gears, rolling bearings, loads, load spectra, lubricants, materials and tools.

In [22] a manual load path oriented housing construction is designed to show the lightweight potential of a car transmission housing. The integration of automatic topology optimization into a GBDL is described by [24] and also makes sense for the application to a gearbox housing. A gear-box housing which is adapted to a two-stage wheel set transmission respecting certain limits to variable requirements is presented in [12], the realization takes place via a parametric CAD housing model. The topological change of the wheel set, i.e. a change in the number of stages or the replacement of a spur gear by a bevel gear, is not possible in this case.

For the mapping of entire design processes have, among others [31] and [10] successfully applied design languages. A potential optimization based on multiple domains can generally be considered as an advantage of graph-based design languages and is described in the cited sources such as [31]. A multi-criteria optimization using the example of transmissions for electric vehicles shows [11]. It can be concluded that several components for a digital process are already existing, but that a holistic digital process for a large number of product variants is not yet existing.

3 Methodology

The developed method for the design synthesis of gearboxes including the design process, the underlying data model, the gearset arrangement process and the geometry representation of the wheels, shafts and housing is explained in this section.

---

[5] All these program systems have in common to represent only the $n+1$-th aspect of a gear system in form of $n+1$-th program. A comprehensive integration platform for a comprehensive *holistic* modeling and simulation of gear systems cannot be offered.

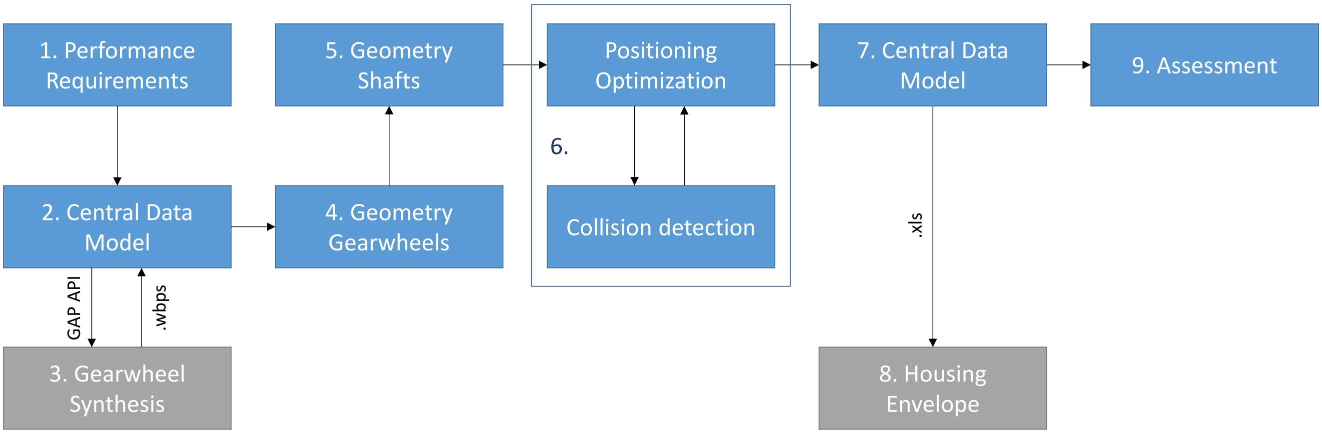

Fig. 2 Flow in the graph-based design language

3.1 Design Process

For the design of gear systems, a process is implemented in a GBDL with which gear system requirements and product requirements can be used to create a rough design of a gear system, consisting of the gearset and the housing. The Design process shown in Fig. 2 starts with the specification of performance data of the product or use case (gear ratio i, engine speed n, engine torque T) and a geometric description of the design space (as .stp, or abstractly described in the form of geometric primitives [28]). The performance data serve as the basis for a call of the system GAP [21] via the existing API that creates a gearset (Fig. 2; Step 2 and 3). In the system GAP a gearset structure may be synthesized, whereas the total gear ratio will be automatically distributed on the individual gear stages. This depends on the required gear ratio and the specification of a design criteria[6]. In this case, an explicit dimensioning of the gear geometry takes place[7]. In addition, various operations can be performed via various program modules, such as the calculation of the reaction forces at the bearing points of the gearset. After the gearset synthesis, the resulting model is read, the backwards data exchange is realized using the .wbps format [8]. For the gearset components in the central data model in the form of a graph, a geometry is then modeled via design rules, e.g. the gear wheel or the shaft provided with shaft sections called 'Sectors' are designed (Fig. 2; Step 4 and 5). In this step nearly no limits exist because of the connection with a CAD system. After the geometrical refinement, the initial gearset is determined by a position logic based on [7] and is positioned in three-dimensional space. For this, an optimization algorithm is used (Fig. 2; Step 6). Once the gearset has been positioned, a parametric CAD model is fitted with the parameters of the gearset (Fig. 2; Step 7 and 8). The communication takes place via an excel file in .xls format. Finally, an evaluation of the overall design is possible.

3.2 Data model

The class diagram of the design language models the relationships between the components as well as the connection to the respective application. In class-own operations and in the activity diagram, the knowledge about the design of the components is implemented.

The generated data model makes it possible to map gear sets consisting of any number of spur gear stages and bevel gear stages. Within this chain of stages, the navigation is done automatically during the design process. Thus, commands are available with which the power flow in the transmission path can be traversed from the input shaft to the output shaft. Each shaft has two bearings and one or two gears. Gears can be spur gears or bevel gears. The implemented operations for components may vary depending on the characteristics, e.g. express the geometry of the component. A simplified graph of a single stage gear system is shown in Fig. 3. As a result of the operations mentioned, instances of the geometric representations in the CAD system are also created for the abstract component instances ('GearWheel' gets a Revol, 'Sector' of a 'Shaft' gets a Cylinder etc.). A realization in the available CAD systems is possible at the push of a button.

---

[6] Specifications: only spur gear stages (SP), bevel gear stage (BE) and spur gear stages, required safety margins, desired compromise in the target conflict of moment of inertia, gearset mass and utilization of strength; complete information in [21].

[7] For spur gears according to the design method of Niemann / Winter [19] or according to a method derived from DIN 3990 [4] / ISO 6336 [13]. For bevel gear and hypoid stages, the design is carried out in accordance with ISO 23509 Annex B [15] or by a method according to FVA 411 [33].

[8] Data format of the FVA Workbench, thus manually generated gearsets can be imported

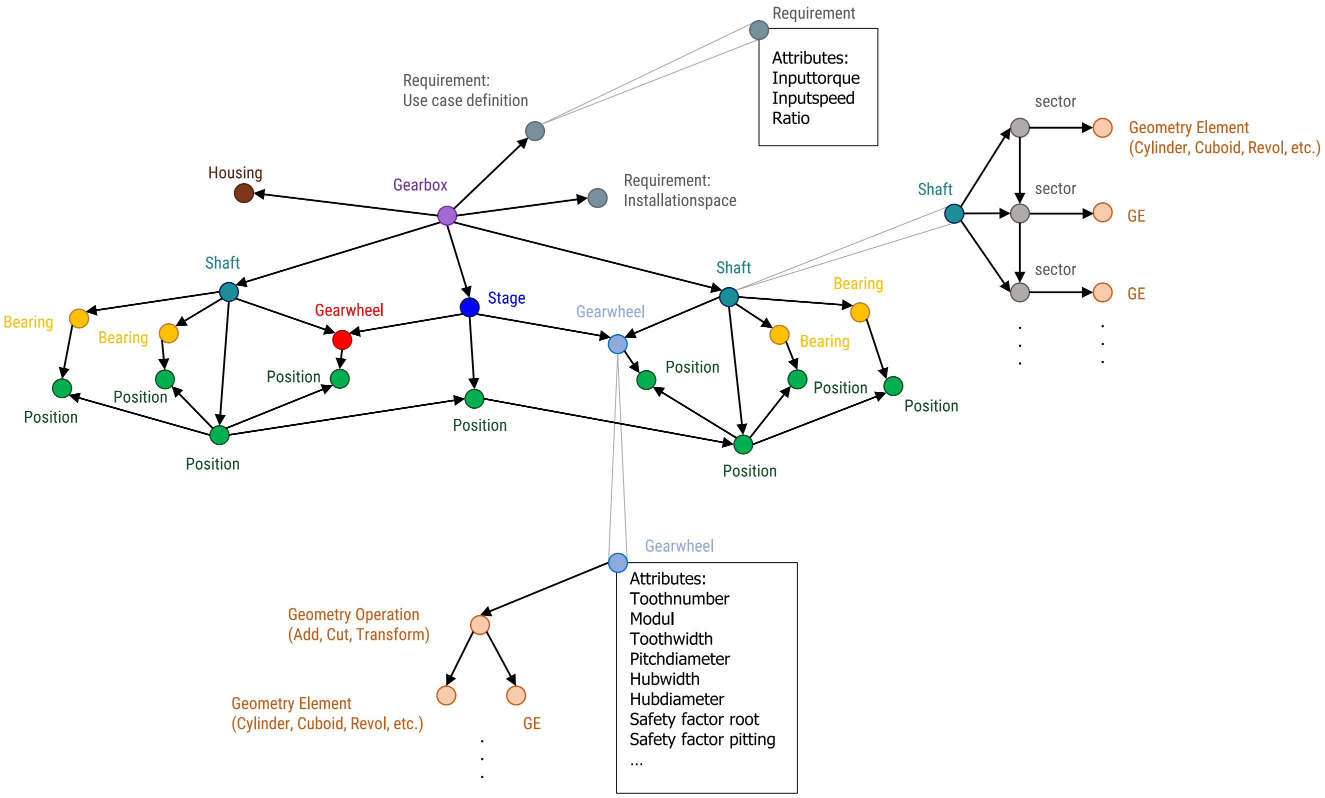

Fig. 3 Data representation as graph, here specific design variant for a single stage gearbox

3.3 Arrangement process in three dimensional space

3.3.1 Position algorithm

The present design system is equipped with a position logic (Fig. 2; Step 6) which could position any gearset schemes consisting of spur gears stages and bevel gear stages to reach given input and output positions in predefined space. This makes it possible to apply very quickly transmission schemes to a spatial problem and to carry out the evaluation against the desired properties of the gearset early in the design process. For the mentioned positioning logic of the gearset, an approach from robotics using modified Denarvit Hartenberg parameters is used, which interprets the parameters of individual gear stages as geometric transformations. The method was first introduced in [7]. As an example a simple spur gear stage may be kinematically interpreted as follows: the length of the first shaft as displacement L, the radial center distance as displacement a and the direction of the second shaft as ϕ (see Fig. 4).

Each gear stage kinematically represents a coordinate transformation with the transformation matrix $[T]_{NJ-1}^{NJ}$. These transformation matrices contain displacements and rotations and can be calculated from the step parameters in Fig. 4. A series of stages (from $i = 1 \dots NJ$) in a gearset can thus be seen as a multiplication of transformation matrices (formula 1). If the position of the output shaft is to be determined, a matrix multiplication of all transformation matrices must be carried out. The result is the matrix $[T]_0^{NJ}$ where the position of the last element of the chain in space can be read out, which correspond to the position of the last shaft in the gearset scheme.

$$[T]_0^{NJ} = [T]_0^1 \cdot [T]_1^2 \cdots [T]_{NJ-1}^{NJ} \tag{1}$$

If a request for a power transmission in space is designated (each a vector for drive shaft and output shaft), then it is necessary to minimize the deviation between the position of the output shaft from the initially given transmission structure (determined by GAP) and the target position. That leads to an optimization problem, where the deviation between the target position and the initial position of the last stage element (output shaft) can be described by an objective function. In order to achieve a predetermined spatial position of the output shaft in space, the transformation matrices and thus the parameters of the stages are changed by an optimization algorithm so that the formulated target function is minimal. In Fig. 5 at the top and bottom two different gearset structures are shown, where on the left the initial structure resulting from the system GAP and on the

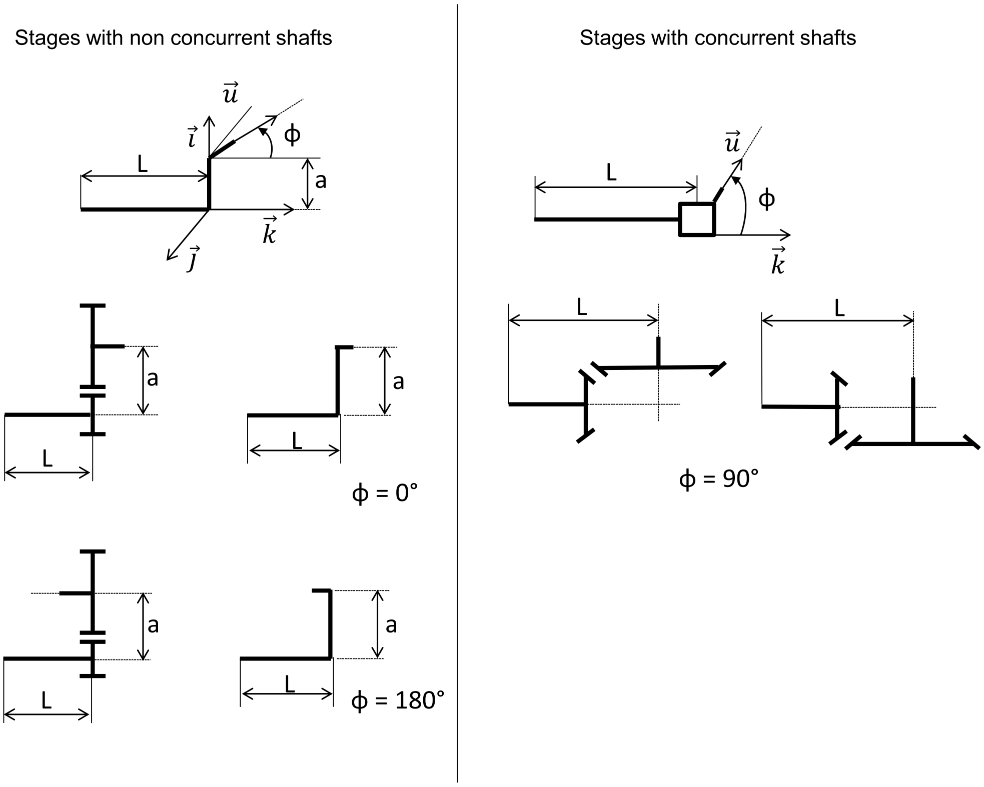

Fig. 4 Parameters for spur and bevel gear stages [7]

right the same gearset structure with changed parameters and reached target position can be recognized.

In advance, the user has to determine which parameters are variable within the optimization process. There are parameters that change the gear main geometry (spur gear: axle center distance a; bevel gear: Gear pitch angle Φ) and thus require a new gear design and the parameters that only affect the arrangement, but not the gear geometry. These include the swivel angles of the stages around the shafts and the lengths of the shafts between the gears. The objective function describes the total length of the prismatic connections q_i of the gearset, i.e. the total length of the shafts and center distances[9] in the transmission. In contrast to [7] the term 'total gear path length' has added a term for the sum of the total mass of the gearset components m_i. This allows the optimizer not only to achieve short and most direct possible paths, whereas axial distances are bridged through thin (fast rotating) shafts at the transmission input sides. Furthermore long high torque shafts at the gear output will be avoided[10]. The optimizer minimizes the objective function 2.

$$F(q_1, \cdots q_{NJ}) = \sum_{i=1}^{NJ} q_i + \sum_{i=1}^{NJ} m_i \qquad (2)$$

For imagination the first term of objective function (formula 2) correspond to a 'rubber band', which is pulled through the gearset skeleton and contracts the gearset in the optimization process on the shortest possible path from the entrance to the exit. The optimizer is a nonlinear gradient-free optimization algorithm[11].

3.3.2 Collision detection

Taking into account the objective function, there are proposed gearset configurations, which individual components of the gearset penetrate each other. A collision check of the components during the runtime of the positioning represents a sensible solution to this problem. For this, the graph of the gearset is traversed at the beginning of the positioning and for each component (gear, bearing, shaft) it is determined against which other gearset components a collision is inadmissible. This information is stored in the form of a matrix and is checked during each optimizer iteration[12]. Since CAD systems are available for the implementation of

---

[9] if center distances are enabled as optimization variables

[10] and vice versa for transmissions with a ratio i < 1

[11] 'fminsearch' from the Optimization Toolbox of Matlab or alternatively the 'PowellOptimizer' from the Apache Commons Math libray with comparable results

[12] A gear wheel does not have to be checked against its partner wheel, likewise all bearings and gears do not have to be checked against their own connected shaft

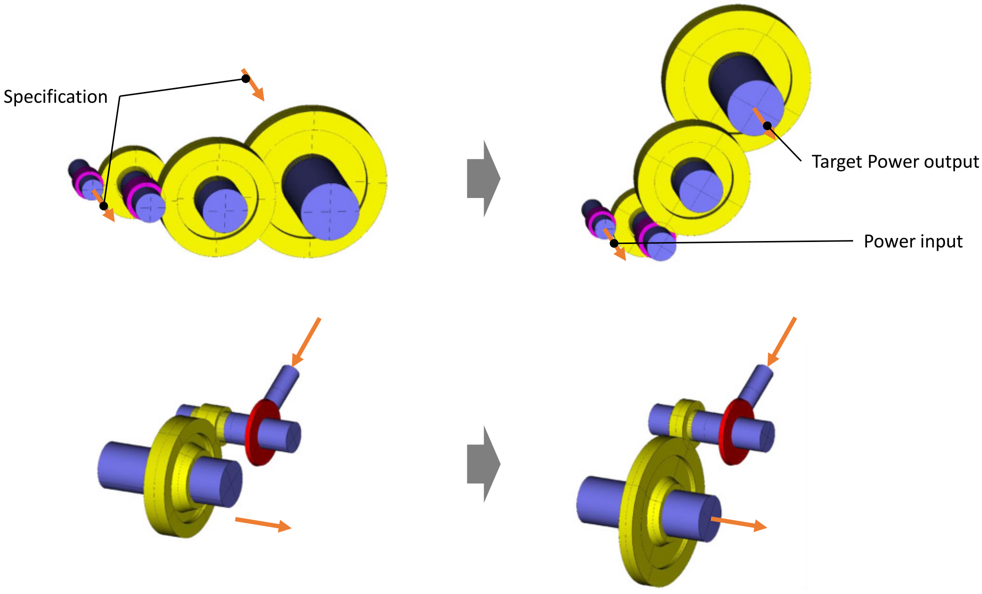

Fig. 5 *Top*: Three stage spur gear drive; *Bottom*: A bevel gear stage and a spur gear stage; *Left*: Before optimization process; *Right*: After optimization process

the geometry in the design system, the collision check can be performed via a CAD kernel (e.g. OpenCascade); the number of penetrations (or the volume of these) can also be added to the target function and must be minimized. The penetration test is currently being implemented; as a temporary solution the penetration is limited by predetermined minimum lengths of shafts, but cannot be prevented e.g. for wheel bodies. Also, the bearings are currently positioned by a simple logic with a minimum clearance next to the gears.

3.4 Geometry representation

3.4.1 Wheel and shaft design

The design of the wheel body is based on [19]. Depending on the wheel dimensions and the ratio between shaft diameter and gear root diameter, there are different implementations of the geometry of the gear body in the design process. As an example, a solid cylindrical design or a waisted wheel body with hub can be applied. The geometry generation of a shaft starts with an analysis on how many components the shaft has to carry on, or rather how many components are linked. Based on this information, the shaft gets automatically divided into shaft sectors, with its sector instances (Fig. 3, right). That lead to a dependence of the shaft dimensions (within specific shaft sectors) to the respective component. Thus the shafts will be get parametrically defined (e.g. diameter and length). For sectors between components, an average diameter of the surrounding components is determined, as an alternative a larger diameter could be created that can serve as a contact surface. Because the geometry is described in abstract form of geometric primitives and linking operation of those (cut, add, transform), this geometry can be expressed via the existing interfaces in CAD systems. Currently available are OpenCascade, Siemens NX and Catia.

3.4.2 Parametric Housing

Another key point in the gearbox design process is the creation of the gearbox housing geometry. The housing design process may begin immediately after the gearset design has been established. As a part of the design automation presented in this paper, which focuses on the early phase of product development, a parametric housing model is currently available and connected [1]. Different gearset schemes require topologically different housings with more or fewer bearing seats and holes for shafts. Topological changes are hard to implement and not robust in parametric CAD models, so in this approach for each relevant standard gearset parametric housing CAD models are once prepared. Currently a suitable housing for up to three spur gear stages can be generated. In the GBDL, after the gearset is defined and arranged, the relevant housing parameters are determined and used for the housing parameterization. Because of the non-abstract formulation of the housing geometry a data exchange with the CAD system takes place via an Excel file with the variable parameters. With such a CAD model a very quick response to changing gearset dimensions within certain limits, for a specific topological gearset is possible. In Fig. 6, such a housing for two-stage gearsets can be identified, which has 32 independent design parameters. Exemplary three parameters are varied here: together, the tooth facewidth of the first and second stages; together, the ratio of the two stages via the gear diameter and together, the diameters of the bearing seats. For each of these

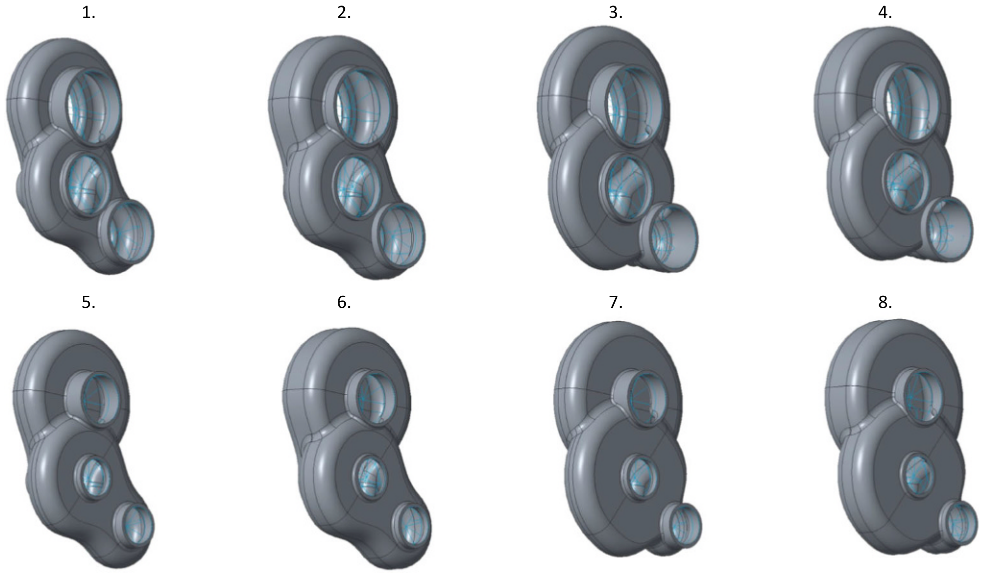

Fig. 6 Parametric housing for a two-stage helical gearbox; Variation of bearing diameters, Gear diameters (gear ratio) and gear width

three parameter combinations two values have been chosen, which will result in the 8 design variants of the same housing model, which are illustrated in Fig. 6. Adding further parameters and varying them independently with more than two assumptions of values, results in a very large number of possible package shapes that are included in such a parametric model.

In principle, in a parametric CAD model the entire functional range of a CAD system can be used to carry out a satisfactory housing design, which leads to detailed and thus high-quality gear housing models. Combined with design variables which are changed via design language a parametric casing model appears to be a promising method. As a limitation for these parametric models, first of all, the domain of definition can be called and therefore the parameter ranges for which a geometry can be generated. Secondly, the compatibility of certain parameter combinations, which also prevents meaningful geometry could be named. As mentioned change of parameters of the housing do not lead to a topological change of the parametric model. Depending on the product requirement, the implemented design language generates different gearset structures, therefore for any desired structure, such a parametric model must be provided once in advance. A limitation of the gearset structures can be derived from practical gear designs, however, this limitation is at the expense of the generic concept of the GBDL and thus limits the design language to a pre-defined specific application.

4 Results and Discussion

The focus of this section are the results of the application of the methodology described in Section 3; these results cover real and synthetic use cases.

4.1 Applications on real use cases

4.1.1 Initial data

The design process and its methods introduced in previous section are applied to three real use cases (UC). Based on the results of three usecases, the method is examined. The usecases correspond to gear systems for electric rail vehicles. In the development of rail vehicle gear systems, the supplier typically gets specifications for the installation space in the form of the surrounding components (bogie, wheel-set axle and electric motor), as well as the performance data of the vehicle and powertrain (torque curve of the electric motor, target gear ratio, axle load, etc.) from the ordering OEM. To approximate this scenario, a space for the gear system has been extracted from the surrounding components, as well as the stub shaft of the electric motor and the axle set in the form of a vector in space. The performance data represents the first instances in the graph of the central data model and serves as the root element of the resulting gear system data model. In Fig. 7, the vectors for gear system input and output can be identified in the left column of the screen (orange) in the installation space derived from the environment.

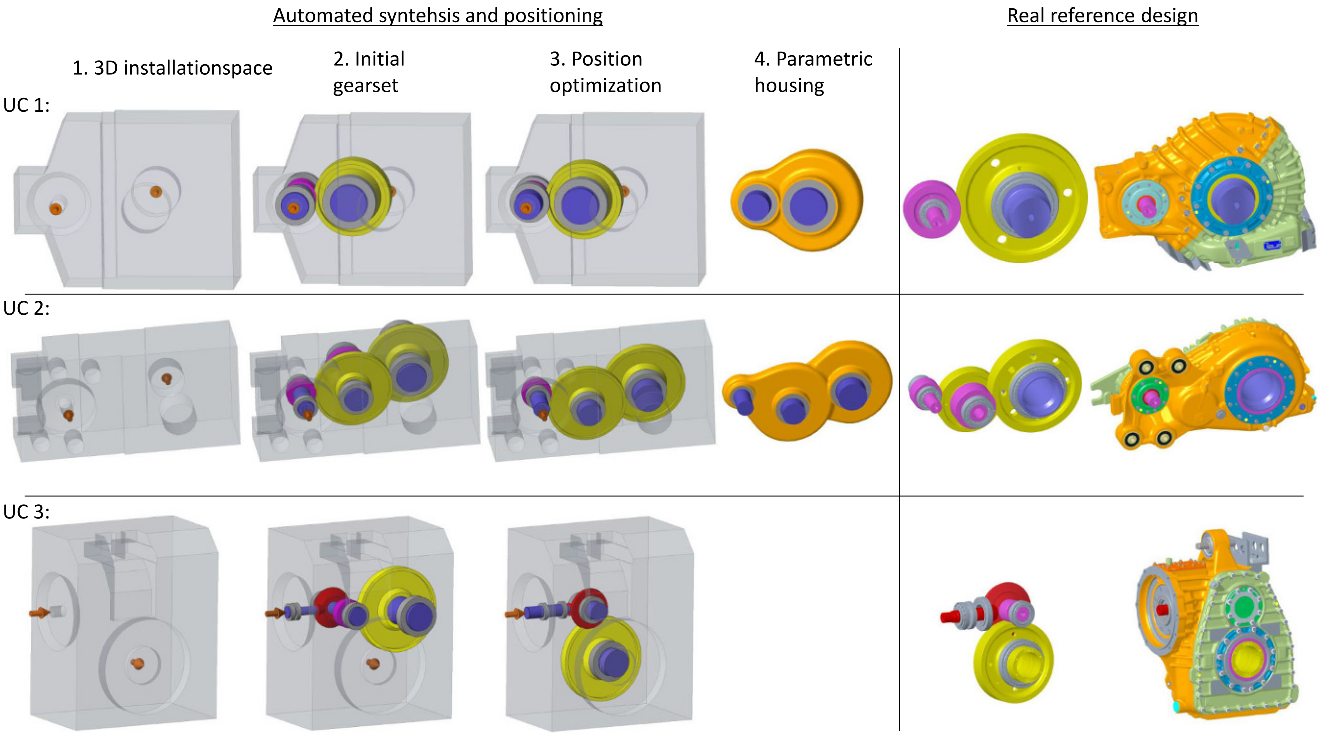

Fig. 7 Usecase 1 to 3; From the requirements to the positioned gearset; Bevel gears in *red*; Spur wheel in solid construction in *pink*; Spur gear with waisted bridge and hub in *yellow*; Shaft in *blue*; Bearings in *gray*

4.1.2 Gearset arrangement results

In the second column of Fig. 7 the initial gearset that is generated using the performance data in the system GAP can be seen, here already with the designed wheel bodies. The third column shows the result of the positioning-algorithm, the gearset with optimized parameters, so that the specified target position in angle and position is approximated. On the far right the respective reference gear system with gearset is shown, from which the application was derived.

Since the gearset design is already completed at the time of positioning, the gearset parameters that relate to the gear geometry, i.e. the axle center distances and gear pitch angles (for bevel gear stages), are fixed. For bevel gear stages, the pivot angles around the respective input shaft of the stage and the shaft length remain free. In the case of spur gear stages the pivot angle and the length of the input shaft are also released. For the shown application examples shown, the system GAP always comes to the same gearset diagram as the reference gear designs. In application case 1, the optimizer can indeed set the correct rotation of the stage to the output vector, because of the fixation of the center distance it does not reach the target position. Accordingly, the housing shell in column four is parameterized for the wheel bodies, which are too small. A possible remedy for gearsets consisting exclusively of spur gear stages is the comparison between the sum of all center distances of the initial gear set with the radial distance between input and output. If the sum is greater than the distance, the gearset can be fitted else the axis spacing must be released as an optimization variable at one or more stages. After the positioning process, a new gear design must be carried out in GAP with the specified center distance for the respective steps with changed center distance in order to obtain a valid gear geometry. As a special rule for single-stage helical gearboxes, the desired center distance must be specified at the beginning for the gearset synthesis in GAP. With regard to the results of the positioning algorithm in case of application 2, it can be noted that the installation space can not be maintained without penetrating the components, see the upper part of Fig. 8.

In addition, a ‘buckling’ of the gearset in downwards direction (as shown for UC 2, optimized) or upward direction is possible, i.e. the function to be minimized has two minimums for this gearset; whether both are found depends on the employed optimization algorithm. It should also be noted that for the control of the GAP design criteria are necessary; with these already a pre-selection is carried out whether the gearset scheme may contain bevel gear stages or not. A wrong choice makes a convergence of the positioning algorithm impossible. In addition, the design criteria ‘minimum mass’ implicitly selects a compact gearset with associated small center distances, which is not necessar-

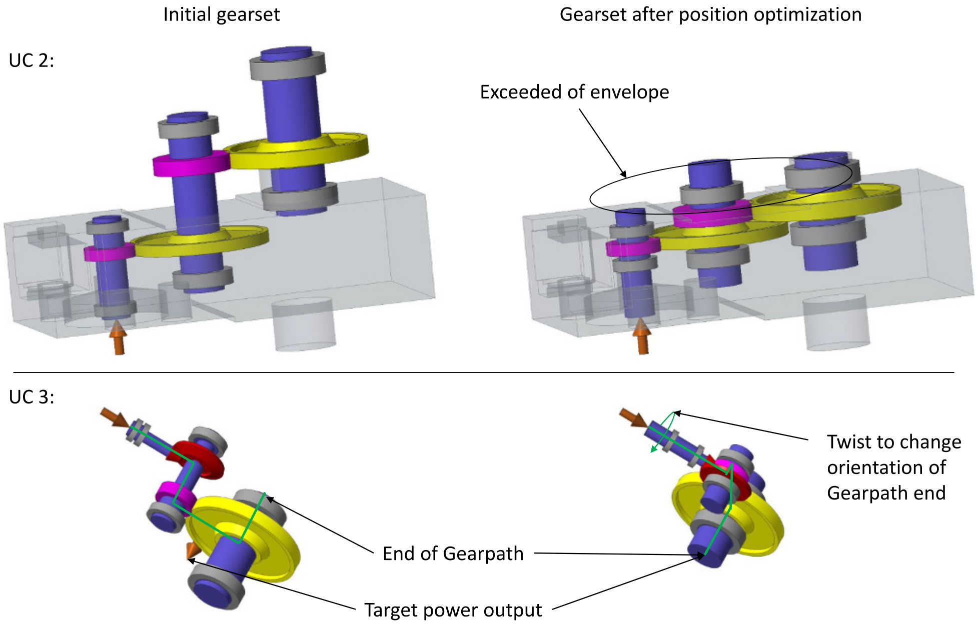

Fig. 8 Usecase 2 out of Envelope; Usecase 3 rotation around inputshaft

ily in accordance with the requirement to bridge a certain spatial distance (see Fig. 7, UC 1).

For application 3, the output is directed forward. Since the orientation of the bevel gears on the shafts and the gear pitch angle are not optimization variables, the optimizer has made a turn around the input shaft. This can be seen in the orientation and positioning of the crown gear – the spur gear is located after the positioning in front of the crown gear of the bevel gear, see the lower part of Fig. 8. In applications in which the gear system output can be anywhere on an axis, the target position must be chosen carefully.

4.1.3 Housing results

For the usecases 1 and 2, which consist exclusively of spur gear stages, the parametric CAD model can be realized as described in section 3. By means of the gearset dimensions, the necessary parameter set for the CAD model is automatically generated, the results are depicted in Fig. 7 in column four. At the time of publication, the housing covers do not match the level of detail of the designs being made. It lacks elementary design features such as mounting points of the gear system or parting planes for the assembly process. However, housing models generated in this way can be used for an estimation of the total gear system mass or a cost estimation. A simple cost estimate seems possible on the basis of the required mold box size for a targeted casting process and the use of materials.

4.2 Application to a syntethic use case

The last use case is a gearset in which the schema is generated manually in GAP[13] and serves to demonstrate the performance of the positioning algorithm. The result of the fourth application is illustrated in Fig. 9. The positioner creates a gearset which reaches the target position in angular position and linear position. As free parameters, the swivel angle of the spur gear stages and bevel gear stages, as well as the output shafts of the spur gear stages are selected. It should be noted that, depending on the target vector (angle and position), also configurations are possible that do not converge.

5 Outlook

The proposed design system is a method that can be used to generate a wheel-set including gear design and spatial positioning of the components as well as a housing shell automatically within seconds of program runtime based on requirements for a powertrain.

For usecases 1 to 3, gearset diagrams are produced which correspond to the manually developed reference gear systems, also the positioning appears comparable. Usecase 4 shows that even spatially complex tasks can be mastered in principle. However, a convergence of the positioner is not necessarily the case, so in configurations with multiple

---

[13] such a structure of bevel gear stages and spur gear stages usually has little practical relevance, accordingly, such a structure is not proposed by the gearset synthesis in GAP

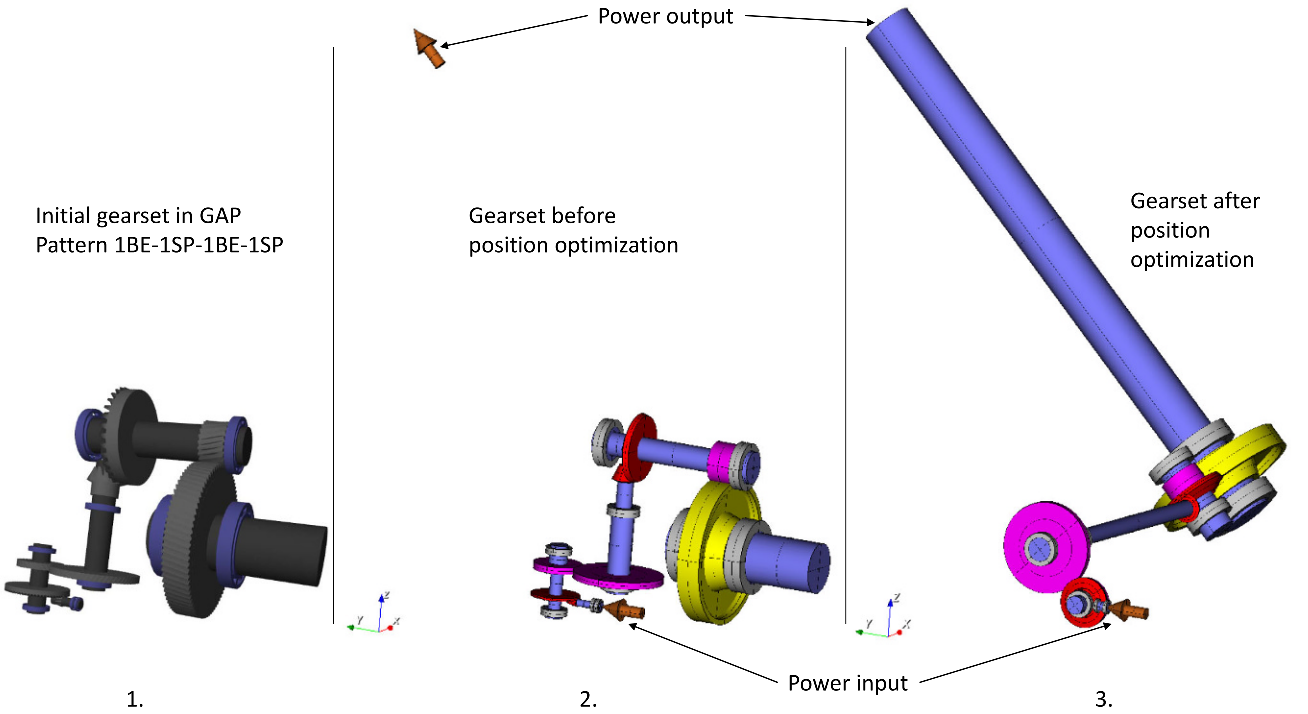

Fig. 9 Uscase 4; Gearset with two bevel gearstages (BE) and two spur gearstages (SP); before and after positioning

bevel gear stages in the gearset, a reaching of the end position and angular position sometimes cannot be achieved. It remains to be investigated, which spatial configurations cannot be mastered at present and whether the choice of an alternative optimization algorithm is expedient. The method of [7] is generally also suitable for hypoid gear stages and worm gear stages, but a use of these stage types is not implemented in the design language at the time of publication. Also, the implementation of a collision detection during optimization iterations is not yet complete. Since requirements concerning the installation space are included only after the definition of the tooth geometry, it can be recognized, for example in application case 1, that with regard to package or spatial bridging, an optimal gearset geometry is not chosen in a straight away. The reason for this is the goal constellation between the compact gearset (GAP, depending on the design criteria) and the desired spatial bridging. Especially with single-stage gearsets, releasing the axle spacing as a variable of the optimization run makes sense, as well as necessarily re-gearing after positioning.

Basically, a productive use of the design system in the early phase of the product development process seems conceivable. With regard to the geometric design of the components, a great deal of development effort remains in implementing corresponding rules for the component characteristics. The implementation provides the participants with requirements regarding the component relevant knowledge and capabilities regarding the abstract implementation in the computer system of the design language. The gear system housing generation with a parametric model can be achieved by the preparation of a model with a high level of detail, but with regard to the use of gearsets with different schemes and thus other housing topologies the limits are quite quickly reached. The method is suitable for a restricted and defined area of application or cases that are taken into account during the implementation. Currently, the system shown is unsuitable to process or propose special cases such as gearsets with power splitters or gearsets with wheel chains. Only current spur gear stages can be taken into account for the housing development in the current implementation; in addition, each shaft of the gearbox needs exactly two bearings located next to the gear wheels. In order not to prefabricate a parametric CAD model for every relevant gearset design, a generic housing model is currently being developed. A skeleton is generated individually on the basis of the gearset, so that the method has fewer restrictions with regard to topology.

The use of a design language solely for automating the design process could lead to a kind of configurator for standard gear systems; this kind of use of a graph-based design language including the link between requirements and functions is described using the example of the design of coffee makers [30].

A linear sequence of automatic design mechanisms does not represent the full potential of a design language. The use of a design language for multi-criteria product optimization in the early phase of the product development process seems particularly useful. For this purpose, the scope of the evaluation of the concept is still to be expanded and part of future research. In principle, all product data can be generated or processed and collected in the form of the design graph, so there is nothing hindering a subsequent evaluation of the design with regard to the relevant domains. A product optimization with regard to several criteria is described in [31]. At the present time, work is being carried out on the automatic evaluation of the resulting gearsets and housings. Evaluation processes for gear systems are e.g. strength analysis of the housing via FEM, efficiency calculation, package and mass inspection, cost accounting and the determination of the gearing safety factors. These processes are frequently carried out manually and are generally accepted. The goal is to evaluate multi-domain product property designs in an automated iterative process to find the best product designs. Ultimately, the target can be the automatic exploration of the design space for gear systems, which can be used to select the proven good gear design from the whole space of possible solutions.

Acknowledgements The project „digital product life-cycle“ (ZaFH, see https://dip.reutlingen-university.de for details) is supported by a grant from the European Regional Development Fund and the Ministry of Science, Research and the Arts of Baden-Württemberg, Germany (information under: www.rwb-efre.baden-wuerttemberg.de).

References

1. Aicher F, Kesenheimer S (2019) Automatisierte Konstruktion eines Getriebegehäuses. Hochschule Ravensburg-Weingarten, Weingarten (unpublished student work)
2. Bae I, Hyong Chong T (eds) (2007) Development of an automatic design algorithm for multi-stage gear drives with applications
3. Bansemir G (2013) Konstruktionsleitsystem für den durchgängig rechnerbasierten Zahnradgetriebeentwurf. Technische Universität München, Diss. Technische Universität München, München
4. DIN Deutsches Institut für Normung: DIN 3990 Tragfähigkeitsberechnung von Stirnrädern (1987)
5. DIN Deutsches Institut für Normung: DIN 3960 Begriffe und Bestimmungsgrößen für Stirnräder (Zylinderräder) und Stirnradpaare (Zylinderradpaare) mit Evolventenverzahnung (1987-03). https://www.beuth.de/de/norm/din-3960/1340388. Accessed: 5 June 2019
6. Dyla A (2002) Modell einer durchgängig rechnerbasierten Produktentwicklung. München, Techn. Univ., Diss. Technische Universität, München
7. Fauroux JC, Sartor M, Paredes M (2000) Using the skeleton method to define a preliminary geometrical model for three-dimensional speed reducers. Eng Comput 16(2):117–130. https://doi.org/10.1007/PL00007189
8. FVA GmbH (2018) FVA Workbench. https://www.fva-service.de/de/software/fva-workbench-50/. Accessed: 5 June 2019
9. FVA GmbH (2018) REXS reusable engineering exchange standard. https://www.rexs.info. Accessed: 5 June 2019
10. Groß J (2014) Aufbau und Einsatz von Entwurfssprachen zur Auslegung von Satelliten. Universität Stuttgart, Diss.. Universitätsbibliothek der Universität Stuttgart, Stuttgart (http://elib.uni-stuttgart.de/opus/volltexte/2014/8971)
11. Hofstetter M (2018) Multi-objective gearbox design optimization for xEV-axle drives under consideration of package restrictions. Forsch Ingenieurwes. https://doi.org/10.1007/s10010-018-0278-9
12. Hofstetter M, Hirz M, Gintzel M, Schmidhofer A (2018) Multi-objective system design synthesis for electric powertrain development. In: Hofstetter M, Hirz M, Gintzel M, Schmidhofe A (eds) 2018 IEEE transportation electrification conference and expo (ITEC). IEEE, Piscataway, pp 286–292 https://doi.org/10.1109/itec.2018.8450113
13. International Organization for Standardization (2006) ISO 6336 Calculation of load capacity of spur and helical gears
14. International Organization for Standardization ISO 6336-1 Tragfähigkeitsberechnung von gerad- und schrägverzahnten Stirnrädern - Teil 1: Grundnorm, Einführung und allgemeine Einflussfaktoren (2006-09). https://www.beuth.de/de/norm/din-iso-1328-1/280561150. Accessed: 5 June 2019
15. International Organization for Standardization (2016) ISO 23509 bevel and hypoid gear geometry. https://www.iso.org/standard/63832.html. Accessed: 5 June 2019
16. Jaroš M (2007) Integration des STEP-Produktmodells in den Getriebeentwicklungsprozess. München, Techn. Univ., Diss 2007. Technische Universität, München
17. KISSsoft AG (2018) KISSsoft. http://www.kisssoft.ch/deutsch/products/kisssoft.php. Accessed: 5 June 2019
18. KISSsoft AG (2018) KISSsys. http://www.kisssoft.ch/deutsch/products/kisssys.php. Accessed: 5 June 2019
19. Niemann G (1989) Maschinenelemente, 2nd edn. Springer, Berlin, Heidelberg, New York
20. Parlow JC (2016) Entwicklung einer Methode zum anforderungsgerechten Entwurf von Stirnradgetrieben. Dissertation. Technische Universität München, München
21. Parlow JC (2016) FVA-Nr. 421 IV Erweiterung Getriebeauslegungsprogramm: Benutzeranletung Getriebeauslegungsprogramm. Forschungsheft. Forschungsvereinigung Antriebstechnik (FVA)
22. Prill T (2013) Beitrag zur Gestaltung von Leichtbau-Getriebegehäusen und deren Abdichtung. Stuttgart, Universität Stuttgart, Diss., 2013. Universitätsbibliothek der Universität Stuttgart, Stuttgart (http://elib.uni-stuttgart.de/opus/volltexte/2013/8415)
23. Ramsaier M, Holder K, Zech A, Stetter R, Rudolph S, Till M (2017) Digital representation of product functions in multicopter design. Proceedings of the 21st International Conference on Engineering Design (ICED17), pp 369–378 (Vol. 1: Resource-Sensitive Design)
24. Ramsaier M, Stetter R, Till M, Rudolph S, Schumacher A (2017) Automatic Definition of density-driven Topology Optimization with graph-based Design Languages. In: Proceedings of the 12th World Congress on Structural and Multidisciplinary Optimisation
25. Rudolph S (2006) Know-how reuse in the conceptual design phase of complex engineering products: or: ‘are you still constructing manually or do you already generate automatically’? In: Tichkiewitch S, Tollenaere M (eds) Proceedings of the 6th International Conference on Integrated Design and Manufacturing in Mechanical Engineering IDMME 2006, Alpexpo-Grenoble, 17.-19. May 2006 AIP-PRIMECA, Grenoble, pp 23–39
26. Golabi S, Fesharaki JJ, Yazdipoor M (2014) Gear train optimization based on minimum volume/weight design. Mech Mach Theory 73:197–217. https://doi.org/10.1016/j.mechmachtheory.2013.11.002 (http://www.sciencedirect.com/science/article/pii/S0094114X13002292)
27. Schaeffler Technologies AG & Co. KG (2019) Bearinx VIP. https://www.schaeffler.de/content.schaeffler.de/de/produkte-und-loesungen/industrie/berechnung-und-beratung/berechnung/bearinx_vip/index.jsp. Accessed: 5 June 2019
28. Schmidt J (2012) Eine Entwurfssprache für Geometrie. Studienarbeit. Universität Stuttgart, Stuttgart
29. Schmidt J (2017) Total engineering automation. https://www.iils.de/#downloads. Accessed: 5 June 2019

30. Tonhäuser C, Rudolph S (2017) Individual coffee maker design using graph-based design languages. In: Gero JS (ed) Design computing and cognition ’16. Springer, Cham, pp 513–533 https://doi.org/10.1007/978-3-319-44989-0_28
31. Vogel S (2016) Über Ordnungsmechanismen im wissensbasierten Entwurf von SCR-Systemen. Universität Stuttgart, Stuttgart https://doi.org/10.18419/opus-8829 (Dissertation)
32. w3.org (2008) Extensible markup language (XML) 1.0 (fifth edition). https://www.w3.org/TR/REC-xml/. Accessed: 5 June 2019
33. Wirth C, Michaelis K (2009) Hypoidtragfähigkeit. Entwicklung eines Berechnungsverfahrens zur Grübchen- und Zahnfußbruchtragfähigkeit von Hypoidrädern: Forschungsvorhaben Nr. 411 I. Abschlussbericht, vol. 887. Forschungsheft. Forschungsvereinigung Antriebstechnik (FVA), Frankfurt